Detecting 802.11 MAC Layer Spoofing Using Received Signal Strength

Yong Sheng[3], Keren Tan[1], Guanling Chen[2], David Kotz[1], Andrew Campbell[1]

[1]Institute for Security Technology Studies, Dartmouth College;
[2]Department of Computer Science, University of Massachusetts Lowell;
[3]Google, Inc. (at Dartmouth ISTS during this work)

***Abstract*— MAC addresses can be easily spoofed in 802.11 wireless LANs. An adversary can exploit this vulnerability to launch a large number of attacks. For example, an attacker may masquerade as a legitimate access point to disrupt network services or to advertise false services, tricking nearby wireless stations. On the other hand, the received signal strength (RSS) is a measurement that is hard to forge arbitrarily and it is highly correlated to the transmitter's location. Assuming the attacker and the victim are separated by a reasonable distance, RSS can be used to differentiate them to detect MAC spoofing, as recently proposed by several researchers.**

By analyzing the RSS pattern of typical 802.11 transmitters in a 3-floor building covered by 20 air monitors, we observed that the RSS readings followed a mixture of multiple Gaussian distributions. We discovered that this phenomenon was mainly due to *antenna diversity*, a widely-adopted technique to improve the stability and robustness of wireless connectivity. This observation renders existing approaches ineffective because they assume a single RSS source. We propose an approach based on Gaussian mixture models, building RSS profiles for spoofing detection. Experiments on the same testbed show that our method is robust against antenna diversity and significantly outperforms existing approaches. At a 3% false positive rate, we detect 73.4%, 89.6% and 97.8% of attacks using the three proposed algorithms, based on local statistics of a single AM, combining local results from AMs, and global multi-AM detection, respectively.

I. Introduction

It is easy to spoof MAC addresses in IEEE 802.11 wireless LANs using publicly available tools [1], making it possible to implement several 802.11 attacks with commodity hardware. For example, an attacker can masquerade as a legitimate access point to disrupt network connections (for denial-of-service attacks), or to advertise false services to nearby wireless stations (for man-in-the-middle attacks). Existing 802.11 security techniques, such as Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA), or 802.11i (WPA2), can only protect data frames. An attacker can still spoof management or control frames to inflict significant damages (Section II-A). Although IEEE 802.11 community has realized this problem and IEEE 802.11w is underway, given the large number of legacy devices, MAC-layer spoofing imposes a serious threat to wireless networks, which are increasingly central to mission-critical enterprise networks.

In this paper we set out to discover MAC spoofing using only "air monitors" (AMs), off-the-shelf 802.11 devices used to passively sniff wireless traffic, without cooperation from access points (APs) or client stations. Most spoof-detection methods focus on the MAC-layer headers, because they are independent of higher-level protocols and not encrypted while MAC-layer encryption is only applied to the payload.

The analysis of sequence number (SN) field in the MAC-layer frame headers is a common method for spoofing detection [2], which assumes that a legitimate device produces a linear sequence of sequence numbers, and that an attacker cannot easily manipulate its own sequence numbers to match, because of firmware in the network cards. Since the SN counters in the attacker's and victim's cards are likely different, any abnormal SN gaps within the frame sequence from the same MAC address suggests a spoofing attack.

However, some open-source drivers and reverse-engineered firmware allow per-frame SN manipulation, and some MAC-layer frames do not have SN at all, thus invalidating both assumptions of SN-based detection. Ultimately, all MAC-layer header fields may be spoofed [3]. On the other hand, physical-layer information is inherent to radio characteristics and the physical environment, making it much harder to forge and it may be used to differentiate devices. Hall et al. uses the frequency-domain patterns of the transient portion of radio-frequency (RF) signals, as a fingerprint, to uniquely identify a transceiver [4]. This approach requires RF sampling at a rate comparable to the base frequency of RF carrier wave, and thus is demanding on the performance of both the wireless measurement device, such as a RF spectrum analyzer, and the analysis device. This requirement limits its application at scale.

Another approach, recently proposed by several researchers, is to use received signal strength (RSS) to distinguish wireless devices for spoofing detection. RSS is the signal strength of a received frame measured at the receiver's antenna. Many commercial 802.11 chipsets provide per-frame RSS measurements. RSS is correlated to the transmission power, the distance between the transmitter and the receiver, and the radio environment because of multi-path and absorption effects. Typically, a wireless device does not often change its transmission power, so a drastic change in RSS measurements of received frames from the same MAC address suggests a possible spoofing attack. The farther the attacker is from its victim, the more likely their RSS patterns differ significantly and the easier it is to detect the spoofing attacks. With a dense array of AMs, even if an attacker can somehow manipulate its transmission power to mimic the RSS pattern of the victim to one AM, it is inherently difficult to fool the majority of these

AMs, each of which have different radio environment. Faria and Cheriton [5], Madory [6], and Chen et al. [7] proposed different MAC spoofing detection methods, all using RSS measurements with some positive detection results.

We have, however, found that these RSS-based detection methods are not effective due to recent advances in wireless hardware. We conducted a series of large-scale experimental studies of RSS measurements on a testbed that covers our 3-floor building with 20 AMs. Each AM is equipped with two Atheros AR5212 802.11a/b/g radios, providing per-frame RSS readings through two integrated omni-directional antennas. An AM is an embedded device and may not capture all frames sent by transmitters in its range, due to limited resources. Our own AM sniffing software, *basset*, passively captures wireless frames and forwards the key frame features to a centralized *merger*, which removes duplicates and synchronizes timestamps to construct a more complete and coherent frame sequence that is stored for further analysis [8].

We were surprised by our initial results. Although the RSS readings of a given transmitter/AM pair sometimes fit a Gaussian distribution, it was not rare to see multiple peaks in the RSS distributions of other pairs, suggesting that those distributions were a mixture of multiple Gaussian distributions. We discovered that this multi-modal phenomenon is caused by *antenna diversity*, a RF communication technique that is widely adopted by most of 802.11 chipsets and drivers to increase the reliability and stability of wireless connectivity. The difference between the mean RSS caused by two antennas can be more than 5 dB in 20% of cases, or 10dB in 4% of cases. If most of the frames are transmitted through one antenna, or the difference between the two peaks is small, however, the RSS distribution is still close to a single Gaussian. This observation directly invalidates the single Gaussian assumption made by Chen *et al.* [7]. It may also significantly impact the detection accuracy of the methods proposed by Faria and Cheriton [5], and Madory [6], since their work did not consider this effect.

Motivated by this observation, we propose to represent the RSS readings for any given transmitter/AM pair as a Gaussian Mixture Model (GMM) [9]. We developed a RSS-profiling algorithm based on the Expectation-Maximization (EM) learning algorithm for GMMs. Once the RSS profile is established for a transmitter in normal conditions, any significant difference in the RSS patterns is considered as a potential spoofing attack. We then used a likelihood ratio test as a local detection algorithm at each AM. With a hypothesis that coordination among multiple AMs increases detection accuracy, we also developed two global detection algorithms. The first algorithm simply combines local statistics from multiple AMs. The second algorithm works on the frame sequence output by the merger. We show that at a 3% false-positive rate, even the local detection algorithm can detect 73.4% of spoofing attacks, in cases where the attack intensity (the ratio of attack frames to total frames) is greater than 10%. The coordination of multiple AMs can improve the detection accuracy to 89.6% for the first algorithm, and 97.8% for the second algorithm, at the same false-positive rate. We also re-implement the algorithms proposed by Chen [7] and Faria [5]. Our results (Section V-C) show the GMM-based global detection significantly outperforms the existing algorithms.

In this paper, we make three main contributions. First, we discover that antenna diversity is the major cause of multi-modal RSS patterns; second, we present a new GMM profiling algorithm; and third, we compare our approach to two other published algorithms in a live testbed, with better results.

We organize the rest of the paper as follows. We survey 802.11 spoofing-based attacks and related detection methods in Section II. We then describe the key observation regarding multi-modal RSS distributions caused by antenna diversity in Section III. We outline our GMM-based method for RSS profiling in Section IV, and the detection algorithms with experimental results in Section V. We discuss the results, potential applications and possible countermeasures in Section VI, and conclude in Section VII.

II. MAC Spoofing and Related Work

In this section we first describe some 802.11 attacks that are based on MAC-layer spoofing, and we derive the general attack model and list our assumptions. We then survey related methods for spoofing detection.

A. 802.11 Spoofing Based Attacks

A variety of 802.11 misbehaviors are based on MAC spoofing, some of which are benign to other users. For example, the spoofer may want to use a randomly generated MAC address to hide their presence, or to masquerade as an authorized MAC address to circumvent AP's MAC address access-control list [1]. Our focus, however, is on spoofing-based denial-of-service (DoS) attacks, misbehaviors that impact other users by denying or degrading their network services.

Deauthentication/Disassociation DoS [1], [10]: The IEEE 802.11 standard requires a two-step handshake before a wireless station (STA) can associate with an AP. When a STA is associated with an AP, the attacker can send a Deauthentication frame using the forged MAC address of the AP. The STA becomes disassociated and has to associate with the AP again. By continuously sending such spoofed Deauthentication frames, the attacker can break the wireless connectivity between the STA and the AP. Note that the attacker may also forge these frames using STA's MAC address.

Power-saving DoS [10]: A STA in 802.11 networks may enter a sleeping state to conserve energy, and its associated AP buffers any inbound traffic for that STA. The attacker can send a PS-Poll frame to the AP by masquerading the STA, then the AP sends the buffered frames and discard them. These frames, however, are lost because the victim STA is still in sleeping state. The attacker may also forge AP's beacons to prevent a STA entering its sleep state, quickly draining its battery.

To successfully launch above mentioned DoS attacks, i.e., to continuously damage the victim, the attacker needs to send out enough forged frames. Bellardo [10] injects forged Deauthentication/Disassociation frames at 10 frames per second (fps). We observed that, to completely block both downlink/uplink TCP and UDP traffic, injection rate of over 20 fps was needed.

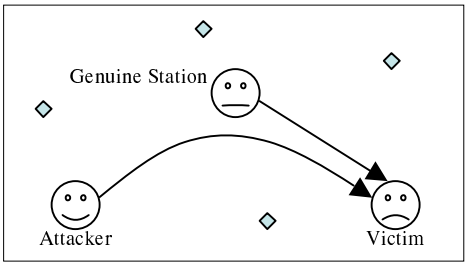

Fig. 1. The roles involved in a 802.11 MAC-layer spoofing

B. Attack Model

In general, the MAC spoofing attack we consider involves an attacker, a genuine station whose MAC address is cloned by the attacker, and a victim who regards the attacker as the genuine station, as shown in Figure 1. A spoofing attack includes two steps. First the attacker uses 802.11 frame manipulation tools to generate the forged frames and then sends them to air using 802.11 frame injection tools. To detect attacks, we deploy an array of AMs (shown as diamonds in Figure 1) to measure the RSS of frames that can be heard at AM's antenna.

We first assume that both the attacker and the genuine station are using off-the-shelf hardware, which means that they use standard 802.11 chipsets as their transceivers. We do not assume anything about their antennas, i.e., the antennas could be integral or external, omni or directional. We further assume that sophisticated attackers may manipulate arbitrary field of 802.11 frames, such as the source and destination MAC addresses, BSSID, ESSID, sequence number, frame checksum, and so on. For each frame the attackers transmit, they may change antenna, power, and bit rate. The attacker may move freely within the area covered by AMs, which implies that an attacker could be close to the genuine station. We also assume that an attacker needs to send enough forged frames to cause damage as discussed in previous subsection. The frames, however, can be injected at any rate.

Our method profiles genuine stations in advance; we assume that attacks are not present during profiling. We assume that the genuine station sends sufficient frames during the profiling period; if necessary, we may send *ping* or *RARP* requests to solicit enough frames. We recognize that the AMs may not capture all frames; AMs often miss frames in practice, due to the AMs' constrained resources, to bursty network traffic, and to collisions in the air. Finally, we assume that the genuine station has a fixed location, which is fortunately true for a common spoof target: production APs. (We discuss the implication of this assumption in Section VI.)

C. Spoofing Detection Methods

We discuss three categories of spoofing detection methods in this subsection. We do not list some heuristics-based approaches, such as counting Deauth/Disassoc frames [10], because they are narrow in focus and can be easily evaded.

1) Sequence-number analysis: The MAC header of every 802.11 management and data frame has a sequence number (SN). The driver or firmware should increase the SN for every new outgoing data or management frame, as required by the standard. When both the attacker and the genuine station are sending frames, an AM who can hear both of them will see SN values from the same MAC address oscillating with two SN sequences in the sniffed trace: one is from the attacker, and the other is from the genuine station. Furthermore, many APs can implement multiple "virtual APs" on one AP, advertising multiple SSIDs; the Aruba Networks APs at Dartmouth are configured with three or more SSIDs, and yet the AP uses a single internal counter for generating sequence numbers. Large gaps might be visible in the sequence numbers transmitted by any one virtual AP. Wright proposes to use these SN gaps as the detection clue [1]. If the gap exceeds a certain threshold, a spoofing alert is raised. This method, however, may raise false alerts in the presence of lost or duplicated frames, which are common in practice. Guo and Chiueh extend this method to use ARP to confirm the current SN from the genuine station, thus reducing false positives [2].

The SN-based approach, however, does not work when the genuine station is silent. Sophisticated attackers may also deliberately forge the SN to evade detection. This approach is also limited by the absence of SN in 802.11 control frames.

2) Transceiver fingerprinting: Every radio transceiver has unique physical characteristics, which lead to unique patterns in the RF signals it transmits. Hall et al. propose to identify a transceiver and thus detect spoofing using *transceiverprints* [4]. They use a wavelet transform to extract frequency-domain features of the transient portion of RF signals, and use fuzzy neural networks to determine whether a given signal matches the profiles or not.

This RF pattern cannot be manipulated at the software level, and is hard to forge by even using a customizable transceiver, such as a software radio. Thus this approach is potentially the most reliable method for detecting spoofing attacks. Profiling the transceiverprints, however, requires sampling the RF signals at a rate comparable to base frequency of the RF carrier (2.4GHz for 802.11b/g, and 5.8GHz for 802.11a). This requirement translates to a higher cost in both measurement and analysis devices, and thus limits its use at scale.

3) Signal-strength analysis: RSS represents the transmission power minus signal attenuation, which is correlated to both the environmental conditions and the distance between the transmitter and the AM. Assuming the attacker and the genuine station are separated by a reasonable distance, RSS can differentiate them and help us detect MAC spoofing. Since signal attenuation often differs significantly from its theoretical expectation, due to many environmental factors, most existing detection approaches rely on statistical methods, or an array of AMs to improve accuracy.

Madory proposed signal strength Fourier analysis (SSFA) for spoofing detection [6]. SSFA is based on the assumption that RSS values from one transmitter follow a fairly tight distribution, while during spoofing attacks the RSS values are interleaved from multiple sources. The coexistence of the attacker and the genuine station cause the RSS values fre-

quently switch between the two sources, resulting in stronger and unpredictable high-frequency components, from the signal processing point of view. SSFA first applies a short-term Fourier transform (STFT) to the RSS values measured by one AM in a fixed frame count window, then calculates the energy of high-frequency components in the frequency domain. An alert is raised if the energy is higher than a threshold. SSFA is a light-weight online algorithm and works even if only one AM is available. It is, however, difficult to improve accuracy by combining RSS values measured by multiple AMs. It may generate many false alerts if the one-source assumption is broken, as we discuss in Section III.

Faria and Cheriton propose to detect spoofing attacks using a *signalprint*, which is the vector of median RSS for a MAC address measured at multiple AMs [5]. To eliminate the effects of transmission power, they actually use the *differential signal strength*, the difference between a median RSS at one AM and the maximum median sensed by all AMs for this MAC address. They propose that two given signalprints represents two transmitters, if the median RSS values measured by at least one AM differ by 10 dB or more. They demonstrated above 95% detection accuracy in their testbed. False positive rate is not reported. They did observe some missing RSS measurements for AMs, and for signalprint-matching they propose to ignore any AMs with missing RSS values. They also occasionally observed strong signal strength oscillation (> 25 dB) for some locations in their experiments, which are similar to the multi-modal phenomenon we discuss in Section III. However, they did not use statistical methods which may improve detection accuracy.

Chen et al. propose a method for detecting spoofing attacks *and* locating the adversary, in both 802.11 WLANs and 802.15.4 ZigBee networks [7]. They assume that RSS values follow a Gaussian distribution with a uniform 5 dB standard derivation. They represent the RSS of a frame measured at N landmarks as a N-dimensional vector, then use the K-mean algorithm to cluster M such vectors (representing M frames sent by a given MAC address) to K clusters. Ideally, each cluster should represent a real transmitter. Assuming $K = 2$, the Euclidean distance between centroids of the two clusters is used for spoofing detection. For 802.11 WLANs, they used a partially-synthetic data set and obtained detection accuracy of 99.2% at a 3.5% false positive rate. In a realistic deployment, however, their algorithm may not work well, as we demonstrate in Section V, due to non-Gaussian RSS distribution and missing RSS measurements. Their work, however, does show that per-frame RSS analysis and multiple-AM coordination are promising for spoofing detection.

Our approach also uses RSS measurements and capitalizes on multi-AM measurements to significantly outperform existing detection methods. We use a Gaussian mixture model to profile RSS patterns, to address the multi-modal RSS distribution caused by *antenna diversity*. Like Faria's work, we also build a normal profile for a transmitter, and detect spoofing attacks by matching to the profiles. Our detector works even if the genuine station is quiet or absent, or there are multiple attackers. Unlike Faria, our algorithm uses per-frame RSS measurements and multiple AMs. We re-implemented Faria's and Chen's algorithms, to the best of our understanding, and compare them below with our algorithm on the same data set collected from a live testbed.

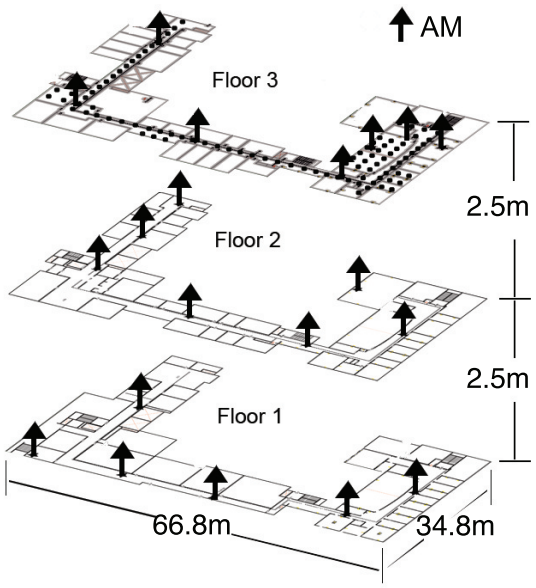

Fig. 2. Our testbed consists of 20 Aruba AP70 AMs (arrows), covering $1,600m^2$ usable space in a 3-floor building. On the third floor, we chose 91 locations (dots), approximately two meters between adjacent locations, to conduct our experiments.

III. RSS Pattern and Antenna Diversity

In this section we first describe our experimental testbed. Then we present the multi-modal RSS pattern observed on one transmitter/AM pair. We further introduce the popular antenna diversity technique and its application in 802.11 WLANs, followed by experimental results demonstrating that antenna diversity is the major reason for the multi-modal RSS pattern.

A. The Testbed

As shown in Figure 2, our testbed is deployed in the Computer Science Department building at Dartmouth College. This 3-floor, $1,600m^2$ office building includes 19 production Aruba AP52 access points (not shown) that provide 802.11a/b/g service to over 80 faculty, staff, and students.

We deployed 20 Aruba AP70 AMs (arrows); each has two 802.11a/b/g interfaces. Each interface contains an Atheros AR5212 chipset, which can provide a *received signal strength indication* (RSSI) for each frame it receives, at 1 dBm granularity in the range $[-100, -35]$ dBm. The AP70 has dual integral dipole (omni) antennas, that are parallel and 5-in (12.5 cm) apart. In our experiments we use only one interface, so that the dual integral antennas fully supports diversity, i.e., the interface may freely switch to either antenna to transmit or receive frames. We reprogram the AP70s with *OpenWRT Linux* (Kamikaze branch, r5494) OS and *MadWifi* (v0.9.2) device driver. We further ran our own AM software, *basset*, to capture wireless frames through *libpcap* (v0.9.5) and to extract

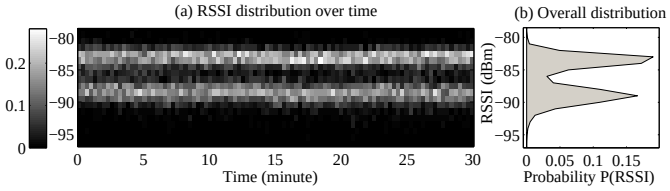

Fig. 3. An example of multi-modal RSS distribution for a transmitter-AM pair. These plots show the RSS distribution of 12,399 frames sent by a production AP in 30 minutes. The stable mixture Gaussian distribution suggests that RSS samples are from two active and stable sources.

features (including physical properties, like RSSI, and MAC-layer header fields). *Basset* forwards the key features of each frame to a server through Ethernet for analysis and storage.

B. Multi-modal RSS Patterns

Many researchers have reported that the RSS distribution for a transmitter-AM pair is approximately Gaussian, but not always accurate. For example, Ladd et al. report that some RSS distributions are essentially non-Gaussian [11]. To study the RSS pattern, we used the 20 AMs to sample RSS from all the frames (on channel 11) in the building. Surprisingly we found that the non-Gaussian distributions were not rare, especially for APs. For example, Figure 3 shows the distribution of 12,399 RSS samples in a continuous 30-minute period, for frames sent by a production AP. The RSS samples in this figure follow a mixture of two Gaussians that are similar in derivations, but have about 6 dB difference in means. Furthermore, this mixture is quite stable over time (30 minutes). This result means that RSS samples are from two active and stable sources, and is not likely caused by temporary multi-path fading or environmental changes.

C. Antenna Diversity

Antenna diversity is a widely adopted technique to improve the quality of wireless connectivity by automatically choosing the best of multiple antennas for receiving and transmitting frames. It exploits the known fact that two antennas spaced a few wavelengths apart (a wavelength is 12.5cm for 2.4GHz signals) have different reception conditions due to reflections or fading. Indeed, most modern 802.11 devices have two (or more) antennas to support diversity.[1]

We thus hypothesize that the RSS samples for a pair of Rx (receive) and Tx (transmit) antennas follows a Gaussian distribution, and the mixed Gaussian distribution we observed is caused by the fact that frames are actually transmitted from and received at the multiple pairs of antennas, in an interleaved manner. This hypothesis is partially endorsed by the MadWifi development group [12]. On the receiver's side, the chipset automatically chooses the antenna on which it detects stronger signal strength of the *preamble* part of a 802.11 frame. On the transmitter's side, there are two cases. For unicast frames going to a given recipient, the driver software initially chooses

[1]Most APs have two or more external antennas. Modern laptops typically integrate two dipole antennas on each side of their LCD screen. In some devices like PCMCIA cards and USB dongles, the antennas are implemented on the printed circuit board (PCB), so they are not easy to see from outside.

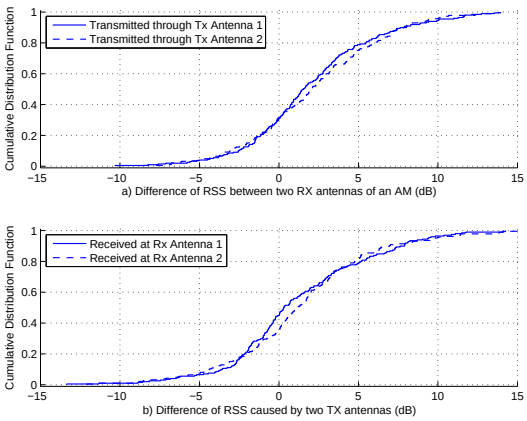

Fig. 4. The difference in RSS caused by antenna diversity.

an antenna to transmit, and sticks with that antenna until recent frames *received* from the recipient have a stronger signal strength on the other antenna. For broadcast and multicast frames, however, the driver alternates antennas. This explains why the multi-modal RSS patterns are more often observed for APs which has more broadcasting fames.

To verify this hypothesis, we used a laptop[2] to send broadcast frames through its two diversity-supporting integral antennas. We used the closest AM as the RSS measurement device, and disabled its Rx (antenna) diversity. We observed multi-modal Gaussian distribution when Tx diversity was enabled on the laptop, and single-modal Gaussian when disabled.

We then moved the laptop to 21 different locations to conduct further experiments; these locations are some of the dots (not shown) marked on the third floor in Figure 2. At each location, we injected 6,000 frames at 100 fps: 3,000 frames through antenna 1, and another 3,000 through antenna 2. We programmed all of the 20 AMs to switch their Rx antenna once per second during the experiment. We extracted an RSSI trace for each combination of (location, AM, Tx Antenna, Rx Antenna), discarding traces with fewer than 50 frames. None of the total 806 traces showed apparent multi-modal distributions. This result suggests that the temporary changes in RSS caused by multiple path fading and other environmental factors be not significant in longer period.

In addition, we calculated the difference of mean RSSI between the two Rx antennas for every triplet (location, AM, Tx antenna), as well as the difference between two Tx antennas for every (location, AM, Rx antenna). The results show that, for either Tx or Rx antennas, the difference in mean RSSI between the two antennas was: a) independent of locations of transmitter; b) independent of locations of the AMs; c) roughly Gaussian; and d) greater than 5 dB in more than 20% cases, or 10 dB in about 4% cases, and could be as high as 15 dB. The cumulative distribution function curves are plotted in Figure 4.

In summary, our results show that antenna diversity is the root cause of the multi-modal RSS distributions. The differ-

[2]IBM Thinkpad T42 with integrated Atheros AR5212 interface, Linux (Fedora 6) and MadWifi (v0.9.3.1).

ence in mean RSSI by using the two antennas for either transmitting or receiving can be high enough to impact the detection accuracy of existing algorithms. In addition, such differences are independent of locations, devices, or the different signal gains/attenuation on each antenna. They are mainly caused by the distance between the two antennas, different orientation, and the multi-path fading. Thus, antenna diversity adds another dimension to the *signalprints* for stationary devices, and thus is even harder for an attacker to forge. On the other hand, appropriately exploiting the multi-modal distributions caused by antenna diversity may actually increase the accuracy of spoof detection. Indeed, a signal-strength approach to spoof detection may be even more effective when the new IEEE 802.11n standard is deployed, as its MIMO technique uses more antennas for transmitting and receiving.

IV. Gaussian Mixture Profiling

We propose to profile the multi-modal RSS patterns using Gaussian mixture models (GMM). We first briefly introduce GMM and the training algorithm, followed by the proposed method and our evaluation results.

A. Gaussian Mixture Models

A Gaussian mixture is defined as a weighted combination of Gaussian distributions [9]. Let x denote a sample scalar value. A Gaussian *pdf* $f(x)$ is parameterized by its mean μ and variance σ,

$$f(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{1}$$

A k-component Gaussian mixture *pdf* $f_k(x)$ is hereafter parameterized by a mean vector $\overline{\mu} = \{\mu_i\}_{1..k}$, a variance vector $\overline{\sigma} = \{\sigma_i\}_{1..k}$, and a positive weight vector $\overline{w} = \{w_i > 0\}_{1..k}$.

$$f_k(x;\overline{\mu},\overline{\sigma},\overline{w}) = \sum_{i=1}^{k} w_i f(x;\mu_i,\sigma_i), \quad \text{where } \sum_{i=1}^{k} w_i = 1. \tag{2}$$

We denote the parameter set as $\theta = (\overline{\mu},\overline{\sigma},\overline{w})$, and write $f(x;\theta) = f_k(x;\overline{\mu},\overline{\sigma},\overline{w})$ concisely. For a given set of n independent samples $\mathbf{x} = \{x_t\}_{1..n}$, the log-likelihood function

$$L(\mathbf{x};\theta) = \frac{1}{n}\sum_{t=1}^{n} \log f(x_t;\theta) \tag{3}$$

measures the goodness that the GMM fits the samples.

Given the initial guesses of parameters $\theta^0 = (\overline{\mu}^0,\overline{\sigma}^0,\overline{w}^0)$, the well-known Expectation-Maximization (EM) learning algorithm [9] efficiently optimizes (locally) parameters that maximize the log-likelihood function, by iterating:

$$g_i^j(x_t) = \frac{w_i f(x_t;\mu_i^j,\sigma_i^j)}{\sum_{l=1}^{k} w_l f(x_t;\mu_l^j,\sigma_l^j)}, \quad \text{for all } i,t \tag{4}$$

$$\mu_i^{j+1} = \frac{\sum_{t=1}^{n} x_t g_i^j(x_t)}{\sum_{t=1}^{n} g_i^j(x_t)}, \quad \text{for all } i \tag{5}$$

$$\sigma_i^{j+1} = \left\{ \frac{\sum_{t=1}^{n} (x_t-\mu_i^j)^2 g_i^j(x_t)}{\sum_{t=1}^{n} g_i^j(x_t)} \right\}^{\frac{1}{2}}, \quad \text{for all } i \tag{6}$$

$$w_i^{j+1} = \frac{1}{n}\sum_{t=1}^{n} g_i^j(x_t), \quad \text{for all } i \tag{7}$$

where j is the number of iterations, being initialized to 0; $g_i^j(x_t)$ is an auxiliary function. The iteration stops when

$$L(\mathbf{x};\theta^{j+1}) - L(\mathbf{x};\theta^j) < \varepsilon, \tag{8}$$

or $j \geq J$, whichever comes first, where ε is a preset small positive number, and J is the maximum number of iterations.

B. Profiling RSS Patterns

We propose to build a GMM profile for each transmitter/AM pair such that the AM can capture enough frames (> 100) from the transmitter. The profiling process can be performed periodically, e.g., once a day or twice a week. During the profiling process, we may send *ping* or RARP requests to some stations, to solicit enough frames, if they are too silent. Once enough RSS samples (say n) are collected for a given transmitter s by AM r, we directly apply the above EM algorithm to train a set of parameters $\theta_{r,s}$ from the n samples. An GMM profile $(r,s,\theta_{r,s})$ is either centrally stored on a server, or on AM r for local detection purpose.

As an infrequent process, we do not care much about the computational costs needed by the profiling process. An EM iteration is $O(nk^2)$ in time. Thus the EM algorithm is bounded by $O(Jnk^2)$, which is still linear in the number of samples. The actual number of iterations varies and depends on ε, initial parameters and training samples.

Choosing the correct number of components, k, is a practical issue. Empirically we chose $k = 2$; although the two Rx and two Tx antennas may lead to 4 distinct Gaussians, it rarely happens as the antenna diversity mechanism automatically chooses the optimal pair to transmit and receive, and the difference between some pairs may be insignificant to observe as an Gaussian component. In addition, using a higher k may lead to *overfitting*, which is harmful for detection.

Another practical issue is to determine the initial parameters. The EM algorithms may converge to a local optimum, depending on the initial parameter set. We randomly choose $k\log_2 n$ pairs of distinct RSSI values in the sample as the initial means, and use a constant 1 dB as the initial σ. The best returning parameter sets are stored as the GMM profile.

Figure 5 shows two examples of GMM profiles to demonstrate how well a GMM profile fits a Gaussian distribution, and a mixture of two Gaussians.

V. Spoofing Detection

In this section we show how to use our GMM profiles for detecting spoofing-based attackers.

A. Single AM

Assume that an AM r captured n RSSI samples $\mathbf{x} = \{x_t\}_{1..n}$ from a MAC address s. Note that all $x_t \in [-100,-35]$ are integers. We now use $p(x;\theta)$ to denote that *probability mass function* of the discrete distribution of $f(x;\theta)$ over its sample space [3]. The spoofing detection is a hypothesis test:

$$\mathscr{H}_0 : \text{the } n \text{ samples } \mathbf{x} \text{ fit the model } \theta_{r,s};$$

[3]The discrete version $p(x;\theta)$ may need to be rescaled from $f(x;\theta)$, such that the sum of $p(x;\theta)$ is 1 for all $x = -100,\ldots,-35$.

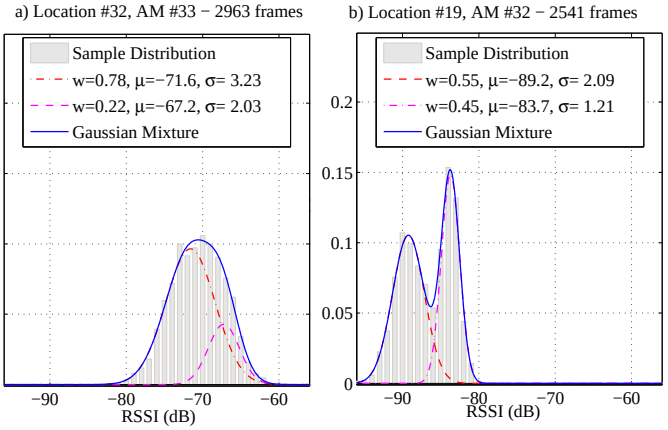

Fig. 5. Examples of GMM profiles.

We slightly abuse the notation of $L(\mathbf{x};\theta)$ to denote the log-likelihood function (3) using the discrete $p(x;\theta)$. Let $h(\theta)$ denote the entropy of the $p(x;\theta)$, given by

$$h(\theta) = -\sum_{x=-100}^{-35} p(x;\theta)\log p(x;\theta). \tag{9}$$

Assuming that $\mathbf{x}$ are sampled over a stationary source independently, it is well known that $L(\mathbf{x};\theta)$ converges, and

$$\lim_{n\to\infty} L(\mathbf{x};\theta) \le -h(\theta), \quad \text{if } \mathbf{x} \text{ is sampled over } \theta'. \tag{10}$$

The equal holds only if $\theta' = \theta$, with probability 1. Thus, $\mathscr{H}_0$ is rejected (i.e., a spoofing attack is detected) if

$$L(\mathbf{x};\theta_{r,s}) + h(\theta_{r,s}) < C_0, \tag{11}$$

where C_0 is a is a model-independent constant threshold we use for detection. This is also known as the *likelihood-ratio* test of a discrete *i.i.d. (independent and identical distribution)*.

In addition, we exploit the limited sample space of RSSI to speed up the calculation of $L(\mathbf{x};\theta)$, by pre-calculating $p(x;\theta)+h(\theta)$ for all integer $x = -100,\ldots,-35$, and storing the values as a vector. Thus the detection algorithm needs only n table look-ups and $n+1$ simple arithmetic operations to calculate (3) for the n RSSI samples. This optimization allows the detector to run on resource-constrained AMs.

B. Multiple AMs

We may obtain better results by using information from multiple AMs. We consider two approaches.

1) Merging local statistics: A straightforward algorithm to merge the results from multiple AMs is to merge their local statistics. Let $\mathbf{r} = \{r_a\}_{1..A}$ denote a set of AMs. Assume that in a time period, AM r_a captures RSSI samples from MAC s as $\mathbf{x}_{r_a,s}$. It calculates its local metric

$$m_a(\mathbf{x}_{r_a,s};\theta_{r_a,s}) = L(\mathbf{x}_{r_a,s};\theta_{r_a,s}) + h(\theta_{r_a,s}), \tag{12}$$

and forwards to a central detector. The central detector then merges the local metrics for s in the same time period, by calculating a global metric[4]

$$M_{\mathbf{r}}(s) = \frac{1}{A}\sum_{a=1}^{A} m_a(\mathbf{x}_{r_a,s};\theta_{r_a,s}), \tag{13}$$

and raises an alert if $M < C_1$.

2) Global detection: It may be more effective to make the global decision using a collated sequence of frames captured by all AMs [8]. Let us assume that MAC address s transmits n frames, denoted as $\mathbf{F} = \{F_t\}_{t=1..n}$. Each frame F_t in the collated sequence is labeled with the set of AMs that heard the frame, and its RSSI measurement $x_{r_a,s}(F_t)$, which for brevity we denote as $x_{a,t}$. If r_a heard F_t, $x_{a,t} \in [-100,-35]$; if not, we denote the missing value as $x_{a,t} = \phi$.

As we discussed in Section II-C.3, a missing RSS reading may be caused by several reasons, and thus it is difficult to find a likelihood function for the missed values. For a given pair of (r,s), we propose to introduce the empirical missing rate $\xi_{r,s}$ (defined as the fractional ratio of missed frames to total frames, during the profiling process) to the GMM profile. Let $p(x;\theta,\xi)$ denote the *pmf* of x for the enlarged sample space,

$$p(x;\theta,\xi) = \begin{cases} (1-\xi)p(x;\theta), & x \ne \phi. \\ \xi_{r,s} & x = \phi, \end{cases} \tag{14}$$

and let $h(x;\theta,\xi)$ denote the entropy of $p(x;\theta,\xi)$. Based on this adjustment, our new global metric is defined as

$$G_{\mathbf{r}}(s) = \frac{1}{n}\sum_{t=1}^{n}\frac{1}{A}\sum_{a=1}^{A}\left(\log p(x_{a,t};\theta_{r_a,s},\xi_{r_a,s}) + h_{r_a,s}\right). \tag{15}$$

The algorithm generates an alert for spoofing if $G < C_2$.

C. Evaluation

We used the same laptop to send 3,000 frames at each of 91 locations (dots on the third floor in Figure 2), while enabling Tx antenna diversity. All the 20 AMs enabled their Rx antenna diversity. For each location, we used the first 1,000 frames as the *training* trace to profile the RSS pattern and the rest 2,000 frames as the *testing* trace for evaluation.

To evaluate the performance of our algorithms in real scenarios, we simulate attack traces for every ordered-pair of locations (s_1,s_2) by mixing the *testing* traces from s_1 (as the genuine station), and from s_2 (as the attacker), by assuming that traces collected from different locations are from different transmitters. We simulated five traces for each (s_1,s_2) pair with attack intensities (AI, defined as the ratio of attack frames to the total frames in a mixed trace) at five different levels: 0%, 1%, 10%, 50%, and 100%, by sampling frames uniformly in time from two testing traces. In our evaluation, we run all the three algorithms against the mixed traces. We treat a trace as a *negative* sample if $AI = 0\%$, or $s_1 = s_2$, or *positive* otherwise.

This method, mixing real traces recorded from one laptop set at different locations, allows us to try far more pairs of locations than practically feasible, reduces potential differences due to changing environmental conditions, avoids any potential bias caused by differences between a genuine laptop and its spoofer, and allows us to run the same traces through all three competing detection algorithms.

We use the receiver operating characteristic (ROC) curves to evaluate detection accuracy of applying an algorithm on a

[4]Here we use the mean, but in future work expect to explore the median or maximum as possible alternatives. Each has its own advantages.

trace set. A point on the ROC curve is a pair of false positive rate (FPR), and detection rate (DR) calculated by applying the algorithm on all traces with a certain threshold value. The ROC curve is then plotted by varying the threshold values.

Figure 6(a) shows the ROC curves for traces whose $AI \geq 10\%$ using the three proposed algorithms: one AM (Section V-A), merging local statistics from 20 AMs (Section V-B.1), and the 20-AM global detection (Section V-B.2). At false positive rate (FPR) 1%, these methods successfully detected attacks in 64.4%, 78.1%, and 94.4% of cases (respectively). The detection accuracy increases to 73.4%, 89.6%, and 97.8% (respectively) when the FPR is 3%. Note that we included every location pair and set $AI \geq 10\%$. The advantage of a global perspective, i.e., analyzing a collated sequence of frames from the merger, is evident in the relative performance of these three approaches.

Figure 6(b) shows the accuracy of our best performing global detection algorithm, under different attack intensities. At 1%-FPR, 91.2% attack traces from all pairs of locations were detected when $AI = 10\%$, and 99.3% when $AI \geq 50\%$. For "trickle" attacks ($AI = 1\%$), the detection accuracywasis extremely low (less than 50% at the 1%-FPR). About 73% of trickle attacks were detected, when FPR was about 20%.

To evaluate the impact of distances over detection accuracy, we set $AI \geq 10\%$ and show the results in Figure 6(c). At 1% FPR, the global algorithm detected 84.3%, 91.0%, 95.5%, and 99.9% attack traces in which the distance between two locations was less than 3m, between 3 and 6m, between 6 and 10m, and greater than 10m, respectively.

We also implemented the detection algorithms proposed by Faria [5] and Chen [7]. We chose the 50% attack intensity level, as the half-to-half mixture should boost the performance of Chen's K-Mean algorithm. In addition, we only used the 7 AMs deployed on the third floor to conduct the comparison, because their studies generally used 4 to 6 AMs (or landmarks) for evaluation. Figure 7 shows that the GMM-based global algorithm using 7 AMs detected 98% attacks at 1% FPR, or 99% at 5% FPR. It significantly outperformed the other two approaches. Faria's signalprints successfully determined more than 70% attack traces with 1% FPR, or 75% at 5% FPR. Note that we used the second max differential DB as the test statistics, as suggested by Faria [5]. Chen's algorithm did not work well on a real data set, due to mixture of multiple sources caused by antenna diversity.

VI. Discussion

Since RSS measurements are dependent on the distance between a transmitter and a receiver, they have often been used for location determination. However, localization and spoof-detection are two different problems. Localization is based on the assumption that all gathered RSS measurements are from a single station and, based on this assumption, the localization algorithm correlates a point in the RSS-measurement space with a point in the physical space. Spoofing detection does not know if all collected RSS measurements are from a single

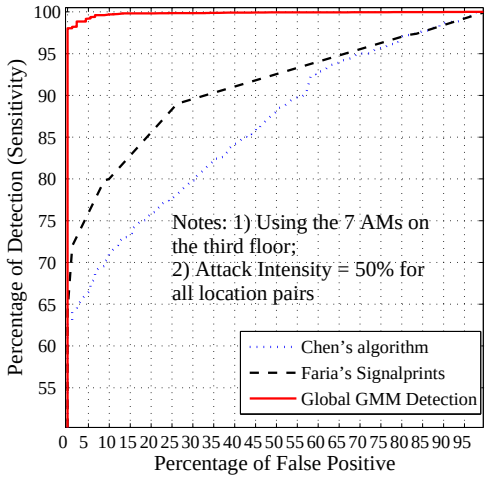

Fig. 7. Comparing GMM with two other approaches.

station, and tries to determine whether they are indeed from the same station.

Because the RSS pattern of a moving station is different from that of a stationary station, in this paper we assume the genuine station is stationary, i.e., this station does not move between profiling events, so we can obtain a stable GMM profile. This assumption works well for production APs and many laptops. An attacker, however, may spoof a high-mobility device, and still be able to inflict damage. One may be able to extend our GMM algorithms to continuously profile RSS patterns; the challenge is to determine whether a deviation of GMM profiles for RSS samples obtained from two adjacent time windows indicates mobility, or a spoofing attack.

We currently assume that the RSS profiles are stable, between profiling events. In our experiments, RSS was stable across our 30-minute measurement periods. Further study is needed to determine how RSS changes and how often profiling may be necessary. Some enterprise-class wireless networks provide automatic reconfiguration of APs, adjusting power levels and channel assignments to optimize coverage while minimizing contention between neighbors. Most such systems reconfigure infrequently, at most once every hour or every day. With clues obtained by monitoring log records from the network-management software — our method can re-compute an AP's profile whenever it is reconfigured.

We currently assume that there are no attacks in progress during profiling. If an attacker were spoofing a genuine station during the profiling period, the RSS profile is polluted with two transmitters. Subsequently, our method would raise numerous alarms, especially when the attacker stops or moves, because the genuine station's behavior no longer fits the profile. After investigating the situation, eliminating the attacker if necessary, the system can re-profile the station.

We assume that a sophisticated attacker may change its transmission power, antenna configuration, or bit rate, for its spoofing effort to be more believable. Although our experiments do not evaluate such changes, we note that it would be nearly impossible for the attacker to match the victim's RSS

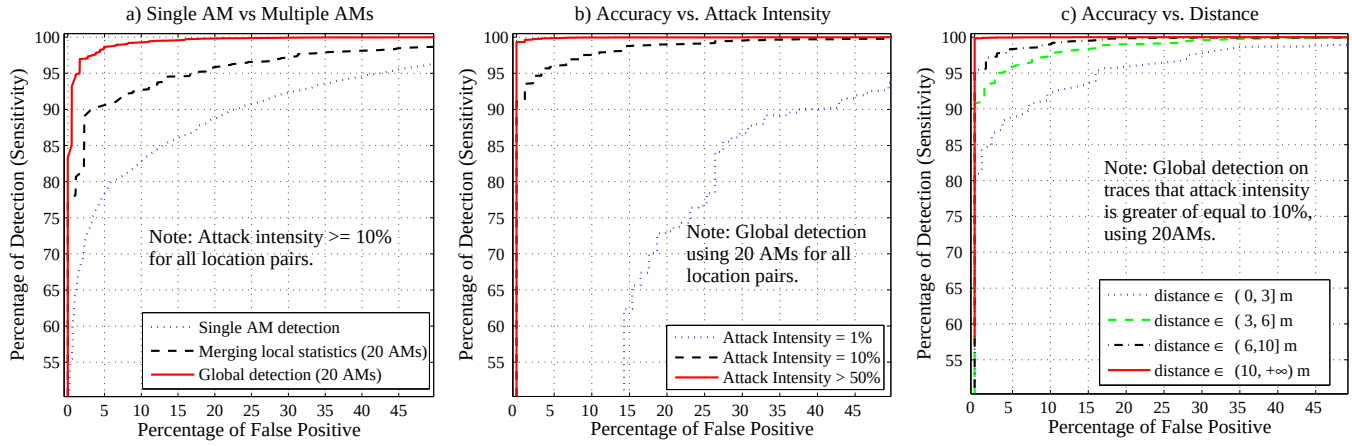

Fig. 6. Receiver operating characteristic (ROC) curves of GMM based spoofing detection.

profile as viewed by multiple AMs, unless it is at the same physical location. A change that makes the attacker sound like the victim from one perspective (AM) will make it seem less like the victim from another perspective (AM). Our multi-AM results clearly show the power of multiple perspectives.

Our experiments show strong evidence for multiple peaks in the RSS histogram, which we determined were the result of antenna diversity in the transmitter and/or receiver. There is a chance, however, that a similar multi-peak histogram could result from a nearby source of interference. The madwifi driver, which we use, actually reports a form of SNR for its RSSI values; interference adds noise, lowering SNR and thus the reported RSSI. If the interference is constant but intermittent, then one might observe two peaks: one high-RSS peak for frames without interference, and one low-RSS peak for frames with interference. We have not observed this phenomenon but it deserves further study.

VII. Conclusion

MAC spoofing attacks in 802.11 networks exploit a fundamental vulnerability of the 802.11 protocol: the MAC addresses of wireless frames can be easily forged, imposing a serious security challenge. Physical-layer information, such as Received Signal Strength (RSS), is hard to forge arbitrarily and can be used to detect such spoofing. Existing RSS-based spoofing detection methods suffer from large RSS variations due to common antenna-diversity technology. In this paper we propose to use Gaussian Mixture Modeling (GMM) for RSS profiling, and show how to use it to detect spoofing attacks. Our detection algorithms, particularly the global decision made by multiple AMs, were very successful and far more accurate than existing approaches, as we have demonstrated using experiments on a building-scale wireless testbed, at least for detecting attackers who spoof the MAC addresses of stationary devices. A key element of future work is to adapt these methods to mobile stations.

Acknowledgments

We gratefully appreciate Sergey Bratus who hypothesized that the multi-modal RSS distribution is related to the dual antennas in most wireless devices. We acknowledge the input and support of the MAP team, including Bennet Vance and Josh Wright, and our other colleagues at Dartmouth College.

This research program is a part of the Institute for Security Technology Studies, supported under award number NBCH2050002 from the U.S. Department of Homeland Security, Science and Technology Directorate. Points of view in this document are those of the authors and do not necessarily represent the official position of the U.S. Department of Homeland Security or the Science and Technology Directorate.

References

[1] J. Wright, "Detecting wireless LAN MAC address spoofing," 2003, technical document. [Online]. Available: http://home.jwu.edu/jwright/papers/wlan-mac-spoof.pdf

[2] F. Guo and T. cker Chiueh, "Sequence number-based MAC address spoof detection," in *Proceedings of the 8th International Symposium on Recent Advances in Intrusion Detection*, Seattle, WA, USA, Sept. 2005.

[3] M. Neufeld, J. Fifield, C. Doerr, A. Sheth, and D. Grunwald, "SoftMAC: Flexible wireless research platform," in *Proceedings of the Fourth Workshop on Hot Topics in Networks*, College Park, MD, Nov. 2005.

[4] J. Hall, M. Bareau, and E. Kranakis, "Using transceiverprints for anomaly based intrusion detection," in *Proceedings of 3rd IASTED, CIIT*, Nov. 2004, pp. 22–24.

[5] D. B. Faria and D. R. Cheriton, "Detecting identity-based attacks in wireless networks using singalprints," in *Proceedings of WiSe'06: ACM Workshop on Wireless Security*, Sept. 2006, pp. 43–52.

[6] D. C. Madory, *New methods of spoof detection in 802.11b wireless networks*. Hanover, NH: M. Eng. Thesis, Dartmouth College, 2006.

[7] Y. Chen, W. Trappe, and R. P. Martin, "Detecting and localizing wireless spoofing attacks," in *SECON'07: Proceedings of the 4th Annual IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, June 2007.

[8] Y. Sheng, G. Chen, K. Tan, U. Deshpande, B. Vance, C. McDonald, H. Yin, T. Henderson, D. Kotz, A. Campbell, and J. Wright, "Securing 802.11 wireless networks through fine-grained measurements," Submitted to IEEE Wireless Communications Magazine.

[9] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Review*, vol. 26, no. 2, pp. 195–239, 1984.

[10] J. Bellardo and S. Savage, "802.11 denial-of-service attacks: Real vulnerabilities and practical solutions," in *Proceedings of the Twelfth USENIX Security Symposium*. Washington, DC, USA: USENIX Association, Aug. 2003, pp. 15–28.

[11] A. M. Ladd, K. E. Bekris, A. Rudys, L. E. Kavraki, D. S. Wallach, and G. Marceau, "Robotics-based location sensing using wireless ethernet," in *MobiCom '02: Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, Sept. 2002, pp. 227–238.

[12] "MadWifi UserDocs: Antenna Diversity," technical document. [Online]. Available: http://madwifi.org/wiki/UserDocs/AntennaDiversity